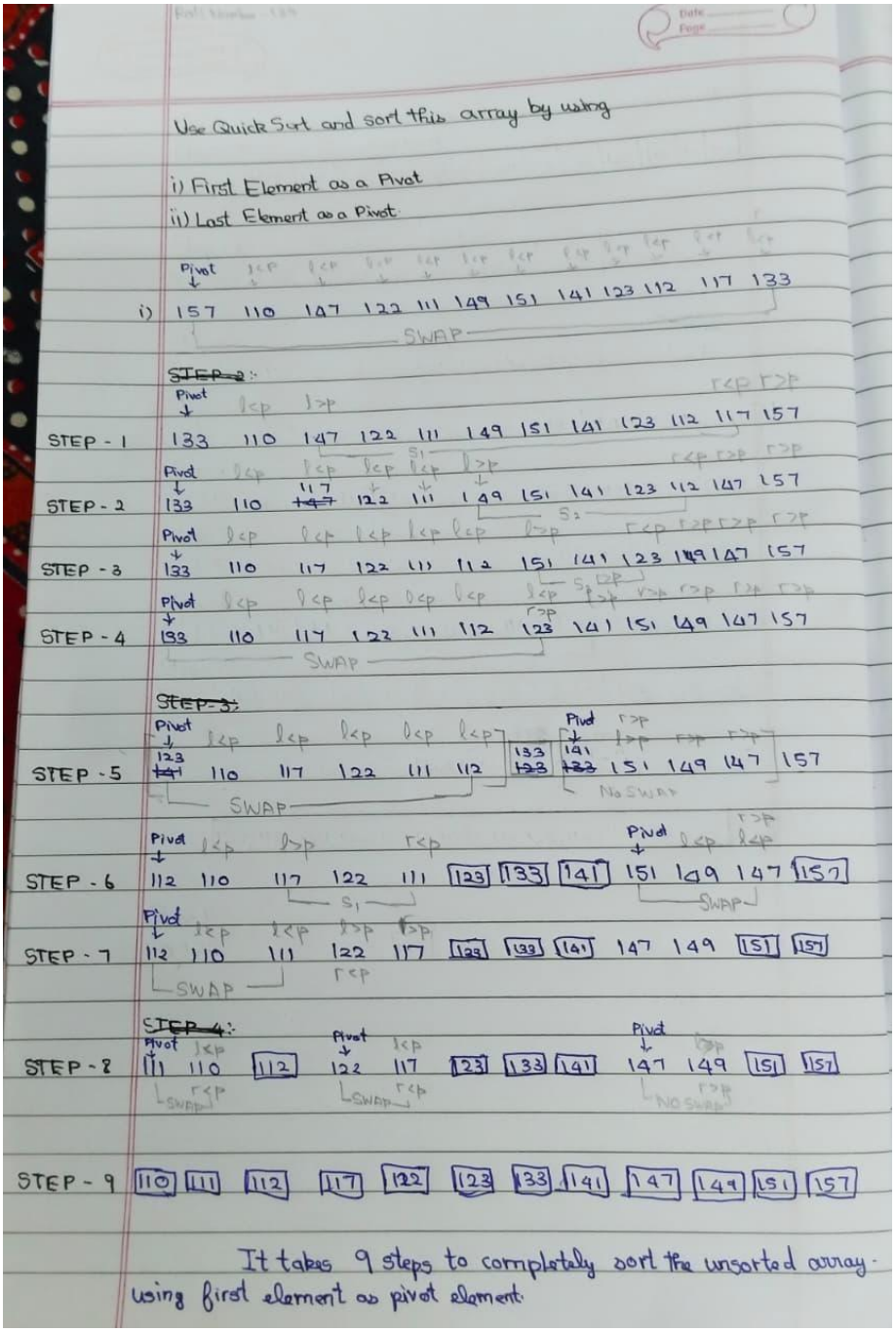
Name : N.ASHWATH

Roll no : CH.SC.U4CSE24133

DAA – Quick Sort Using 3 Conditions

I) First Element as Pivot

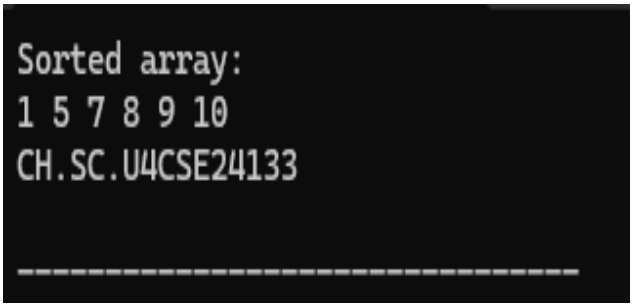
Working :



Program :

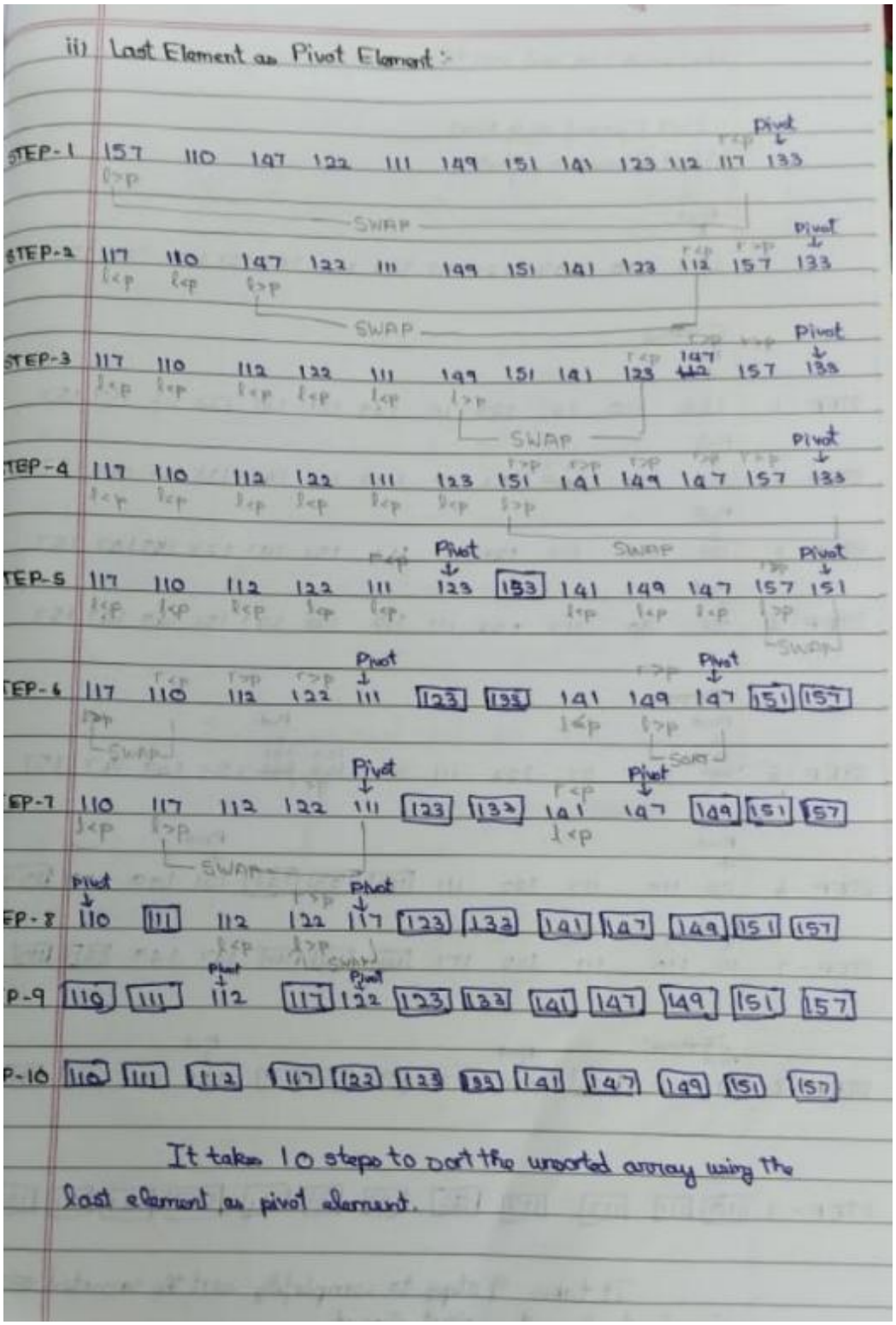
```c
#include <stdio.h>

int partitionFirst(int a[], int low, int high) {
    int pivot = a[low];
    int i = low + 1, j = high, temp;

    while (i <= j) {
        while (i <= high && a[i] <= pivot)
            i++;
        while (a[j] > pivot)
            j--;
        if (i < j) {
            temp = a[i];
            a[i] = a[j];
            a[j] = temp;
        }
    }

    temp = a[low];
    a[low] = a[j];
    a[j] = temp;

    return j;
}

void quickSortFirst(int a[], int low, int high) {
    if (low < high) {
        int p = partitionFirst(a, low, high);
        quickSortFirst(a, low, p - 1);
        quickSortFirst(a, p + 1, high);
    }
}

int main() {
    int a[] = {10, 7, 8, 9, 1, 5};
    int n = 6, i;

    quickSortFirst(a, 0, n - 1);

    printf("Sorted array:\n");
    for (i = 0; i < n; i++)
        printf("%d ", a[i]);

    printf("\nCH.SC.U4CSE24133\n");  // Edited Line

    return 0;
}
```

Output :



II) Last Element as Pivot

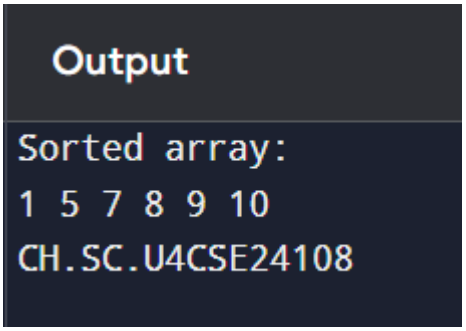
Working :



Program :

```c
#include <stdio.h>
int partitionLast(int a[], int low, int high) {
    int pivot = a[high];
    int i = low - 1, j, temp;
    for (j = low; j < high; j++) {
        if (a[j] <= pivot) {
            i++;
            temp = a[i];
            a[i] = a[j];
            a[j] = temp;
        }
    }
    temp = a[i + 1];
    a[i + 1] = a[high];
    a[high] = temp;

    return i + 1;
}
void quickSortLast(int a[], int low, int high) {
    if (low < high) {
        int p = partitionLast(a, low, high);
        quickSortLast(a, low, p - 1);
        quickSortLast(a, p + 1, high);
    }
}
int main() {
    int a[] = {10, 7, 8, 9, 1, 5};
    int n = 6, i;
    quickSortLast(a, 0, n - 1);
    printf("Sorted array:\n");
    for (i = 0; i < n; i++)
        printf("%d ", a[i]);
    printf("\nCH.SC.U4CSE24108\n");
    return 0;
}
```

Output :



III) Middle Element as Pivot

Program :

```c
#include <stdio.h>
#include <stdlib.h>
int partitionRandom(int a[], int low, int high) {
    int random = low + rand() % (high - low + 1);
    int temp = a[random];
    a[random] = a[high];
    a[high] = temp;
    int pivot = a[high];
    int i = low - 1, j;
    for (j = low; j < high; j++) {
        if (a[j] <= pivot) {
            i++;
            temp = a[i];
            a[i] = a[j];
            a[j] = temp;
        }
    }
    temp = a[i + 1];
    a[i + 1] = a[high];
    a[high] = temp;
    return i + 1;
}
void quickSortRandom(int a[], int low, int high) {
    if (low < high) {
        int p = partitionRandom(a, low, high);
        quickSortRandom(a, low, p - 1);
        quickSortRandom(a, p + 1, high);
    }
}
int main() {
    int a[] = {10, 7, 8, 9, 1, 5};
    int n = 6, i;
    quickSortRandom(a, 0, n - 1);
    printf("Sorted array:\n");
    for (i = 0; i < n; i++)
        printf("%d ", a[i]);
    printf("\nCH.SC.U4CSE24108\n");
    return 0;
}
```

Output :